EXHIBIT B

Initial Release 4.5

Function : Calendaring and Scheduling

SchRetrieve - Retrieve a schedule.

```
#include <schedule.h>

STATUS LNPUBLIC SchRetrieve(
     UNID FAR *pApptUnid,
     TIMEDATE FAR *pApptOrigDate,
     DWORD dwOptions,
     TIMEDATE_PAIR FAR *pInterval,
     LIST FAR *pNames,
     HCNTNR FAR *rethCntnr,
     void FAR *MustBeNull1,
     void FAR *MustBeNull2,
     void FAR* FAR *MustBeNull3);
```

Description :

Synchronously retrieves a local or remote schedule by asking the caller's home server for the schedule.

The ONLY time that local busy time is used is when the client is in the Disconnected mode which is specified through the location document. Otherwise, the API will route ALL lookup requests to the users home server for processing.

Parameters :

input :
pApptUnid - Ignore this UNID in computations.

pApptOrigDate - Reserved. Must be set to NULL.

dwOptions - Option flags:
SCHRQST_COMPOSITE   return composite schedule
SCHRQST_EACHPERSON return each person's schedule
SCHRQST_LOCAL do only local lookup
SCHRQST_FORCEREMOTE force remote even if you are using workstation based email

pInterval - Pointer to a TIMEDATE_PAIR structure that specifies the range over which the free time search should be performed. In typical scheduling applications, this might be a range of 1 day or 5 days.

pNames - Pointer to a list of fully distinguished names whose schedule should be searched. This list is in TEXT_LIST format without the datatype work. This list can be conveniently built with the textlist package.

MustBeNull1 - This parameter must be NULL.

MustBeNull2 - This parameter must be NULL.

MustBeNull3 - This parameter must be NULL.

Output :
(routine) - NOERROR - Successfully retrieved a schedule.
ERR_xxx - There are many possible errors. It is best to use the code in a call to OSLoadString and display/log the error for the user as your default error handling.

rethCntnr - Handle of schedule container results are returned in. If *rethCntnr is NULLHANDLE, then the container will be allocated by this call. If it is not NULLHANDLE then the caller has allocated it and is responsible for freeing it on all errors.

See Sample Program :

MISC\SCHEDULE

See Also :

SchContainer_Free
SchSrvRetrieve

SCHRQST_xxx